

Faculdade de Engenharia da Universidade do Porto



Regras para Suporte à Decisão na Gestão de Projetos

Luís Manuel Paulo Hermenegildo

PARA APRECIÇÃO POR JÚRI

**Relatório realizado no âmbito da Dissertação do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores Major
Automação**

**Orientador: João Pascoal Faria
Responsável por acompanhamento na empresa: Bruno Martins**

Setembro 2014

© Luís Manuel Paulo Hermenegildo, 2014

A Dissertação intitulada

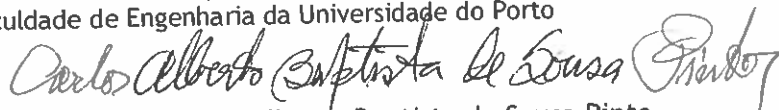
“Regras para Suporte à Decisão na Gestão de Projetos”

foi aprovada em provas realizadas em 02-10-2014

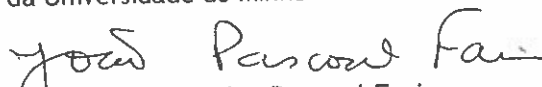
o júri



Presidente Professor Doutor Ricardo Santos Morla
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Carlos Alberto Baptista de Sousa Pinto
Professor Auxiliar do Departamento de Sistemas de Informação da Escola de
Engenharia da Universidade do Minho



Professor Doutor João Carlos Pascoal Faria
Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de
Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Luís Manuel Paulo Hermenegildo

Faculdade de Engenharia da Universidade do Porto

Resumo

A “vida” de uma Empresa é o somatório das suas decisões. Decisões precipitadas podem ser desastrosas, decisões tardias podem significar perdas de oportunidade. Daí ser importante ter uma ferramenta que nos auxilie na tomada de decisão.

O tema da dissertação foi proposto pela StrongStep, que dispõe de um sistema *software as a service* (SaaS), o SCRAIM, para gestão de projetos, incluindo uma vertente para dispositivos móveis (o AMOB).

O SCRAIM ajuda no controlo de projetos, no controlo dos seus impedimentos e riscos, permite verificar se o projeto está atrasado e se este se pode vir a atrasar, facilita também na distribuição de tarefas, permite controlo do projeto durante todo o seu ciclo de vida. O AMOB que é a vertente móvel do sistema, é um complemento ao SCRAIM, permite o acompanhamento do projeto em tempo real, mas é uma ferramenta essencialmente apenas para consulta do projeto.

O objetivo deste trabalho é conceber e implementar um módulo de suporte à decisão, nomeadamente ajudar o utilizador a decidir as ações de tratamento de riscos e impedimentos, para a vertente móvel do SCRAIM (AMOB), sistema esse desenvolvido e comercializado pela StrongStep.

Este módulo permite a um especialista definir um catálogo de ações-padrão para riscos e impedimentos mais comuns; com base num catálogo e num algoritmo de “*matching*”, é capaz de sugerir ao utilizador uma lista priorizada de ações a tomar para tratar riscos ou impedimentos descritos pelo utilizador. O módulo foi desenvolvido em Ruby on Rails (RoR) mas foi também utilizado JavaScript (JS). Como contribuição foi também proposto um catálogo de ações-padrão para os riscos e impedimentos.

Abstract

The “life” of a company is the sum of its decisions. Hasty decisions can be disastrous, late decisions can mean loss of opportunity. Hence it is important to have a tool to assist in decision making.

The dissertation topic was proposed by Strongstep, that has a software as a service (SaaS) system, the SCRAIM, for project management, including an extension for mobile devices (AMOB).

The SCRAIM helps on project control, in control of their risks and impediments, allows you to check if the project is delayed and if the project may ultimately delay, also facilitates the distribution of tasks, allows control of the project throughout its life cycle. The AMOB that is the mobile aspect of the system is a supplement to SCRAIM, allows monitoring the project in real-time, but it is essentially just a tool to query the project.

The objective of this work is to design and implement a decision support module, namely for helping the user to decide the actions of treatment risks and impediments, to the SCRAIM for the shed mobile (AMOB), developed and commercialized by Strongstep.

This module allows an expert to define a catalog of standard actions for the most common risks and impediments; based on a catalog and on a matching algorithm, is able to suggest to the user a prioritized list of actions to take, to deal with risks or impediments described by the user. The module was developed in Ruby on Rails (RoR) but was also used JavaScript (JS). A proposed contribution was also a catalog of standard actions to risks and impediments.

Agradecimentos

Diversas pessoas contribuíram para a realização deste trabalho e a todas elas exprimo o meu reconhecimento e agradecimento. No entanto gostaria de agradecer a algumas delas em particular.

Ao professor João Pascoal Faria e ao Engenheiro Bruno Sousa Martins, por todo apoio e orientação ao longo da dissertação.

Queria também agradecer em especial à minha mãe, ao meu pai, ao meu irmão e à minha namorada, por me terem aturado e estarem sempre do meu lado. Sem eles a realização deste trabalho não seria possível.

Índice

1.Introdução.....	1
1.1 Enquadramento	1
1.2 Motivação	1
1.3 Objetivo	1
1.4 Estrutura do Documento	2
2. Gestão de Riscos e Impedimentos no Contexto de Gestão de Projetos	4
2.1 Gestão de Projetos	4
2.1.1 Conceito de Projeto	4
2.1.2 Objetivos da Gestão de Projetos	4
2.1.3 Atividades.....	5
2.1.4 Variáveis e Áreas de Conhecimento.....	5
2.2 Gestão de Riscos	6
2.2.1 Conceito de Risco.....	6
2.2.2 Objetivos da Gestão de Risco.....	6
2.2.3 Identificação dos Riscos	8
2.2.4 Avaliação dos riscos	8
2.2.5 Tratamento dos riscos.....	9
2.2.6 Tipos de riscos e sua frequência	10
2.3 Gestão de Impedimentos	12
2.3.1 Conceito de Impedimento	12
2.3.2 Processo de Gestão de Impedimentos.....	12
2.3.2.1 Identificação do impedimento.....	13
2.3.2.2 Análise do impedimento	13
2.3.2.3. Avaliação do impedimento	13
2.3.2.4 Implementação de ações	13
2.3.2.5 Controlo e revisão	13
3.Sistemas de Suporte à Decisão	15
3.1 Decisão e processo de decisão	15
3.2 Conceito de Sistema de Suporte à Decisão	16
3.3 Breve História dos Sistemas de Suporte à Decisão	17
3.4 Taxonomias	17
3.5 Arquiteturas	18
3.6 Conclusão	19
4.Concepção e Implementação	21
4.1 Estratégia e Funcionalidades	21
4.2 Arquitetura e Tecnologias	25
4.2.1 Arquitetura	25
4.2.2 Tecnologias	26
4.2.2.1 Ruby on Rails	26
4.2.2.2 JavaScript	27
4.3 Estrutura da Base de Dados	27
4.4 Ficheiros criados e modificados	30
4.5 Algoritmo de Recomendação.....	32
4.5.1 Base do algoritmo desenvolvido	32
4.5.2 Redução das palavras à sua raiz	36
4.5.3 Ordenação das soluções	36
4.5.4 Sistema de tradução	37
4.6 Catálogo de Soluções	37
5.Apresentação da Solução Desenvolvida	40
5.1 Consulta de riscos e ações de mitigação.....	40
5.2 Consulta de impedimentos e ações sugeridas.....	42

6. Conclusões e Trabalho Futuro	45
6.1 Conclusão	45
6.2 Trabalho Futuro	45
Referências	47

Lista de figuras

Figura 1 - Gestão do Risco.....	7
Figura 2 - Gestão de Impedimentos.....	12
Figura 3 - Áreas de conhecimento envolvidas nos Sistemas de Suporte à Decisão.....	16
Figura 4 - Funcionamento geral.....	23
Figura 5 - Diagrama de casos de uso UML referente ao SSD a desenvolver.....	24
Figura 6 - Arquitetura Geral da solução desenvolvida.....	25
Figura 7 - Arquitetura MVC.....	26
Figura 8 - Vista detalhada de todos os riscos.....	40
Figura 9 - Ação para risco selecionado.....	40
Figura 10 - Palavras-chave mais ação para Risco “Missing skills on the team”	41
Figura 11 - Vista detalhada de todos os impedimentos.....	42
Figura 12 - Solução para o impedimento selecionado.....	42
Figura 13 - Palavras-chave mais soluções para o Impedimento “Absent owner holding up key decisions”	42
Figura 14 - Solução introduzida pelo utilizador.....	43

Lista de tabelas

Tabela 1 – Contagem das dimensões dos riscos.....	11
Tabela 2 – Tabela Soluções/Ações.....	28
Tabela 3 – Tabela Issues.....	28
Tabela 4 – Tabela Users.....	30
Tabela 5 – Catálogo de Soluções.....	37

Abreviaturas

Lista de abreviaturas

BI	Business Intelligence
JS	JavaScript
MVC	Model-View-Controller
OLAP	On-line Analytical Processing
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
RoR	Ruby on Rails
SSD	Sistemas de Suporte à Decisão

1.Introdução

A presente dissertação foi desenvolvida no âmbito da unidade curricular de Dissertação do curso Mestrado Integrado em Engenharia Electrotécnica e de Computadores do Major Automação da Faculdade de Engenharia da Universidade do Porto em ambiente empresarial na empresa StrongStep. A dissertação teve como orientador na instituição de ensino o Professor João Pascoal Faria e na empresa acolhedora o Engenheiro Bruno Sousa Martins.

1.1 Enquadramento

A StrongStep é uma empresa localizada no UPTEC - Parque de Ciência e Tecnologia da Universidade do Porto. A Strongstep é uma empresa especializada em engenharia de software que contribui para a melhoria da qualidade de software no mundo. A StrongStep também dispõe de um sistema *software as a service* (SaaS), o SCRAIM, e a sua extensão para dispositivos móveis, o AMOB, que é uma ferramenta para gestão de projetos.

O tema desta dissertação nasce da necessidade que a empresa StrongStep sentiu em melhorar o seu sistema na vertente AMOB. A StrongStep propôs a implementação de um módulo de apoio à decisão capaz de recomendar possíveis soluções/ações para tratamento de problemas (riscos ou impedimentos) indicados pelo utilizador.

1.2 Motivação

A principal motivação para a realização deste trabalho foi a necessidade e a importância que a implementação e desenvolvimento deste módulo tinha para a empresa, aliada ao interesse da minha parte em aprender algo novo abraçando este projeto, neste caso, Ruby on Rails (RoR) e JavaScript (JS).

1.3 Objetivo

O objetivo desta dissertação é a melhoria da ferramenta SCRAIM/AMOB através do desenvolvimento de um módulo de suporte à decisão no contexto da gestão de projetos. Mais

especificamente, pretende-se que o sistema seja capaz de recomendar possíveis soluções (ações de resolução ou mitigação) para problemas (impedimentos ou riscos) indicados pelo utilizador, com base num catálogo de soluções padrão fornecido por especialistas.

1.4 Estrutura do Documento

O presente documento está estruturado segundo no total 6 capítulos. No segundo capítulo é apresentado o resultado da pesquisa feita sobre a gestão de projetos, a gestão de riscos e a gestão de impedimentos. No capítulo seguinte são analisados os sistemas de suporte à decisão. O quarto capítulo refere-se à especificação da solução proposta e sua implementação. No quinto capítulo é apresentada a solução desenvolvida na perspectiva do utilizador. Sendo que no sexto e último capítulo é feita uma reflexão do trabalho desenvolvido e são lançadas propostas para trabalho futuro.

2. Gestão de Riscos e Impedimentos no Contexto de Gestão de Projetos

Neste capítulo serão revistos conceitos importantes de Gestão de Projetos, relevantes para a presente dissertação.

Em seguida é abordada a Gestão dos Riscos, uma vez que parte dos problemas do utilizador são riscos associados ao projeto, revendo também os seus conceitos, bem como as melhores práticas de identificar, lidar e tratar dos riscos associados aos projetos.

Por último é abordada brevemente a Gestão de Impedimentos, visto que parte dos problemas do utilizador são impedimentos.

2.1 Gestão de Projetos

2.1.1 Conceito de Projeto

Segundo o PMI (Project Management Institute) [21], um projeto é um conjunto de atividades temporárias, realizadas em grupo, destinadas a produzir um produto, serviço ou resultado únicos. Um projeto é temporário no sentido de que tem um início e fim definidos no tempo. E um projeto é único no sentido de que não se trata de uma operação de rotina, mas um conjunto específico de operações destinadas a atingir um objetivo em particular.

2.1.2 Objetivos da Gestão de Projetos

“A gestão de projetos é o processo através do qual se aplicam conhecimentos, capacidades, instrumentos e técnicas às actividades do projeto de forma a satisfazer as necessidades e expectativas dos diversos stakeholders que são indivíduos ativamente envolvidos no projeto ou cujo resultado do mesmo poderá afetá-los positivamente ou negativamente” PMI [8].

A gestão de projetos procura também manter o nível de fracasso do projeto num nível tão baixo quanto necessário durante todo o projeto.

De um outro ponto de vista, a gestão de projetos é um alcançar de objectivos ao mesmo tempo que se procura otimizar a utilização dos recursos (dinheiro, pessoas, tempo, etc.), disponíveis para o projeto.

Normalmente a gestão de um projeto é da responsabilidade de apenas um indivíduo. Este não está diretamente ligado ao resultado final, mas antes trabalha para manter o progresso e a interação mútua dos diversos participantes, de modo a reduzir o risco de fracasso do projeto.

Nas últimas décadas emergiram na indústria uma série de abordagens; de entre essas destaca-se a abordagem PMBOK, que se tem tornado uma abordagem padrão na maioria das indústrias.

2.1.3 Atividades

A abordagem tradicional distingue cinco grupos de processos no desenvolvimento de um projeto correspondentes às fases do ciclo de vida dos projetos:

- Iniciação;
- Planeamento;
- Execução;
- Monitorização e Controlo;
- Encerramento.

Para manter o controlo sobre o projeto do início ao fim, um gestor de projetos utiliza várias técnicas, como: Análise de valor agregado, Cronograma, Melhoria de processo, etc.

2.1.4 Variáveis e Áreas de Conhecimento

Aquando da realização do projeto são definidos objetivos e restrições relacionados com diversas variáveis que devem ser cumpridos. A gestão de projetos tenta tomar controlo sobre estas variáveis e garantir que os objetivos a elas associados são cumpridos. As principais variáveis são:

- Tempo;
- Âmbito;
- Custo.

Isto é conhecido também como o “triângulo de restrições”. Qualquer alteração numa destas variáveis vai afetar diretamente as outras. Exemplificando: um orçamento apertado poderia significar o tempo aumentado e o âmbito reduzido; um aumento de âmbito implica

normalmente um aumento de tempo e custo; uma restrição apertada de tempo pode significar custos aumentados e o âmbito reduzido.

De uma forma mais geral, o PMBOK identifica 10 áreas de conhecimento na gestão de projetos:

- Gestão de integração do projeto;
- Gestão do âmbito do projeto;
- Gestão de tempo do projeto;
- Gestão de custos do projeto;
- Gestão da qualidade do projeto;
- Gestão de recursos humanos do projeto;
- Gestão das comunicações do projeto;
- Gestão de riscos do projeto;
- Gestão de aquisições do projeto;
- Gestão de envolvidos do projeto.

Três dessas áreas referem-se precisamente à gestão das 3 variáveis acima referidas. Das restantes 7 áreas, tem particular importância no âmbito da presente dissertação a área de Gestão de Riscos do Projeto, que se detalha a seguir.

2.2 Gestão de Riscos

2.2.1 Conceito de Risco

Segundo o Grande Dicionário de Língua Portuguesa um risco pode ser definido como um perigo, inconveniente, ou fatalidade muito provável de ocorrer, estando o conceito do mesmo associado a uma ideia negativa, mas este tanto pode ter efeitos negativos como positivos.

2.2.2 Objetivos da Gestão de Risco

A gestão do risco não é um processo recente. Ela tem sido utilizada desde sempre, embora de forma pouco sistemática. Os profissionais responsáveis por esta área, socorrem-se da sua prática e gerem os riscos da forma que entendem mais eficaz. Embora possam não estar a utilizar a melhor forma de lidar com o risco, ignorá-lo certamente seria muito pior; a aplicação sistemática da gestão do risco torna os riscos explícitos e obriga a uma descrição formal dos mesmos, o que permite que a sua gestão seja mais eficaz e fácil de conduzir.

Sendo assim, a Gestão de Risco pode ser vista como “Actividades coordenadas para dirigir e controlar uma organização no que respeita ao risco.” [34], ou ainda como, “Processo sistemático de identificação, análise e resposta ao risco. Este tem como objetivo a maximização da probabilidade e consequências dos eventos positivos e a minimização da probabilidade e consequência dos eventos negativos de um projeto.” [35].

A gestão do risco não eliminará os riscos dos projetos, e tem como objetivo assegurar que os riscos são geridos de uma forma mais eficiente.

Segundo Flanagan e Norman [36], existem algumas regras chave na gestão do risco:

- Os riscos devem ser corretamente identificados, classificados e analisados antes de ser considerada qualquer resposta para os mesmos;
- Os riscos que ocorrem efetivamente deixam de ser riscos, passando a ser designados por impedimentos ou incidentes;
- Os riscos não devem ser geridos de forma intuitiva ou sentimental;
- Os riscos devem ser geridos de forma contínua desde o início até ao fim do projeto;
- Os riscos e os seus relatórios devem fluir de forma eficaz ao longo da estrutura de gestão;
- Uma estrutura de risco mal definida produzirá mais riscos;
- Uma visão aberta e uma atitude crítica são os ingredientes chave para prever o que poderá acontecer;
- Um plano de contingência é essencial;
- A gestão do risco não deve ser demasiado complicada nem demasiado simplista;
- Potenciais falhas no projeto devem ser indicadas o mais cedo possível.

São três as etapas que se consideram cruciais na gestão de risco: (1) identificação dos riscos, (2) avaliação dos riscos e (3) tratamento dos riscos (Desenvolver respostas e controlar) (Ver figura 1).



Fig.1 - Gestão do Risco

2.2.3 Identificação dos Riscos

A identificação dos riscos é o processo através do qual se reconhecem os riscos e se definem as suas características essenciais, como o seu contexto, a sua caracterização, e as suas consequências. Esta é a fase mais importante do ciclo, pois se os riscos não forem identificados não podem ser medidos nem controlados, são ignorados.

Esta etapa tem como objetivo gerar uma lista organizada e estruturada dos riscos identificados, sendo que a definição das suas consequências deve ser feita de forma minuciosa.

Existem várias técnicas para uma melhor deteção dos riscos, sendo que duas delas são o *Brainstorming* e a Entrevista Estruturada. O responsável pela gestão dos riscos pode optar por uma técnica ou um conjunto de técnicas, sendo que todas elas têm vantagens e desvantagens. Em seguida será apresentado o *Brainstorming* e a Entrevista Estruturada.

Brainstorming

O *Brainstorming* foi criado pelo publicitário Alex Osborn. É um método que propõe a reunião de um grupo de pessoas para gerar novas ideias acerca de determinado assunto, sendo que nenhuma ideia é considerada errada ou absurda. Para uma sessão de *Brainstorming* ser bem sucedida deve seguir certos parâmetros tais como:

- É proibido a crítica ou debate de ideias. Quantas mais ideias melhor;
- Igualdade de oportunidade. Todos devem ter oportunidade de expor as suas ideias;
- Devem-se também apresentar ideias modificadas ou uma combinação de ideias já propostas, com o objetivo de as melhorar.

Entrevista Estruturada

A Entrevista Estruturada baseia-se numa conversação entre duas ou mais pessoas (o entrevistado e o entrevistador) em que o entrevistador coloca questões ao entrevistado seguindo um guião que estabelece antecipadamente as perguntas a serem formuladas.

Tem como vantagem a fácil análise dos dados, mas em contrapartida não permite flexibilidade nas respostas.

2.2.4 Avaliação dos riscos

O processo de avaliação dos riscos consiste em quantificar e perceber a origem do risco, obtendo um grau de importância para cada risco. Através da combinação das consequências e da probabilidade dos fatores de risco obtém-se o nível de risco.

Devem-se utilizar certas fontes de informação para se efetuar uma avaliação correta das consequências e da probabilidade de cada risco. Como fontes de informação podemos ter as seguintes:

- Registos passados;
- Pesquisa de mercado;
- Experiências práticas;
- Literatura publicada;
- Resultados de consulta pública;
- Experiências e protótipos;
- Modelos económicos, de engenharia ou outros;
- Experiências de especialistas.

A avaliação dos riscos pode ser feita segundo diferentes níveis de detalhe, sendo que, existem três tipos de avaliação dos riscos:

- Análise qualitativa;
- Análise quantitativa;
- Análise semi-quantitativa.

Análise qualitativa, descreve a relevância da probabilidade e das consequências dos riscos através de palavras. Como por exemplo, uma escala poderia ser: insignificante, significativa e prioritário.

Análise quantitativa, descreve a relevância da probabilidade e das consequências dos riscos através da utilização de valores numéricos. Esta caracterização baseia-se em dados registados anteriormente.

A incerteza e a variabilidade das consequências e da probabilidade devem ser tidas em conta na análise quantitativa.

Análise semi-quantitativa, neste tipo de análise são atribuídos valores numéricos a escalas qualitativas. Tem como objetivo produzir escalas mais abrangentes comparando com a análise qualitativa.

2.2.5 Tratamento dos riscos

O tratamento dos riscos é a última fase de todo o processo de gestão do risco, esta envolve as fases desenvolver respostas e controlar.

Existem várias opções de tratamento dos riscos, tais como:

- Evitar o risco, parando temporariamente o projeto ou atividade que origina o risco;
- Aceitar o risco;
- Manter o risco;
- Partilhar o risco com outros através de alterações contratuais, incentivos à produtividade, seguros, garantias.

No tratamento dos riscos é preciso entender que os riscos tratados possivelmente acrescentam novos riscos. Estes, também devem passar por todo o processo de gestão de riscos.

2.2.6 Tipos de riscos e sua frequência

Neste sub-capítulo serão apresentados tipos de riscos e a sua frequência em projetos de software. Os tipos de risco e a sua frequência são diferentes para diferentes tipos de projeto. Foram escolhidos riscos para projetos de software pois a Strongstep está a desenvolver um projeto de software e o SCRAIM/AMOB também é mais focado em projetos de software.

Segundo Wallace [31] existem 6 dimensões de risco: utilizador, requisitos, complexidade do projeto, planeamento e controlo, equipa e ambiente organizacional. Os riscos a seguir apresentados serão categorizados de acordo com estas dimensões.

Segundo Boehm [32] o *top 10* dos riscos em projetos de software compreende:

- Deficiências de pessoal - Equipa;
- Tempo e custo estimado irrealista - Planeamento e controlo;
- Desenvolvimento de funções de software erradas - Requisitos;
- Desenvolvimento errado da *interface* com o utilizador - Requisitos;
- *Gold plating* (Adicionar recursos extra ao produto para encantar o cliente) - Planeamento e controlo;
- Mudanças tardias nos requisitos - Requisitos;
- Deficiências em componentes fornecidos - Planeamento e controlo;
- Deficiências em tarefas realizadas externamente - Planeamento e controlo;
- Deficiências de desempenho em tempo-real - Complexidade do projeto;
- Esforço excessivo das capacidades científicas - Complexidade do projeto.

Já segundo Addison [33] o *top 10* dos riscos compreende:

- Incompreensão das necessidades dos utilizadores - Requisitos;

- Ausência da declaração dos benefícios para o negócio - Requisitos;
- Foco muito estreito sobre os canais de distribuição e negócios em geral - Requisitos;
- Recursos de segurança inadequados que estão a ser construídos no sistema - Complexidade do projeto;
- Falta de experiência no projeto em comércio electrónico - Equipa;
- Falta de conhecimento no design de páginas de internet - Equipa;
- Falta de apoio e comprometimento da alta direção - Ambiente Organizacional;
- Incapacidade de gestão da expectativa final do utilizador - Utilizador;
- Procedimentos insuficientes para garantir a segurança, integridade e disponibilidade da base de dados - Planeamento e controlo;
- Falta de envolvimento e comprometimento do utilizador - Utilizador.

Segundo estes autores as dimensões mais frequentes são o planeamento e controlo e os requisitos (ver Tabela 1).

Tabela 1 - Contagem das dimensões dos riscos

Dimensão	Nº de riscos		
	Boehm	Addison	Total
Requisitos	3	3	6
Planeamento e controlo	4	1	5
Equipa	1	2	3
Complexidade do projeto	2	1	3
Utilizador	0	2	2
Ambiente Organizacional	0	1	1

Isto implica que os gestores de projetos devem ser mais cuidadosos com este tipo de riscos (planeamento e controlo e requisitos), sendo que os de utilizador e ambiente organizacional são menos comuns.

2.3 Gestão de Impedimentos

2.3.1 Conceito de Impedimento

Segundo (PMI, Guia PMBOK, 3ª ed.) um impedimento (*issue*) é “Um ponto ou assunto em discussão ou disputa, ou um ponto ou assunto que não está resolvido e está sob discussão ou sobre o qual existem pontos de vista opostos ou desacordos”.

Apesar de a Gestão de Impedimentos não aparecer explicitamente como área de conhecimento no PMBOK, na prática é normalmente desenvolvida em articulação com a Gestão de Riscos, dada a proximidade dos dois conceitos.

2.3.2 Processo de Gestão de Impedimentos

A diferença entre riscos e impedimentos é que um impedimentos é um risco que acabou por se efetivar. Com uma boa gestão de riscos, o que supõe um plano de mitigação, estes impedimentos têm um impacto menor na organização, pois já saberemos o que fazer para lidar com eles. Se ocorre um impedimento que não estava previsto, a figura 2 resume como lidar com esse impedimento.



Fig.2 - Gestão de Impedimentos

2.3.2.1 Identificação do impedimento

Identificação do impedimento (*issue*) é o processo pelo qual o impedimento é identificado. Os impedimentos podem aparecer de várias fontes, por exemplo, a partir de uma questão levantada por e-mail ou de um problema com o cliente. Assim que um impedimento foi identificado, deve ser registrado no catálogo dos impedimentos.

2.3.2.2 Análise do impedimento

É onde os impedimentos são categorizados por urgência e impacto. Esta classificação permite-nos distinguir quais impedimentos são mais críticos. Um impedimento com um impacto alto, mas com pouca urgência precisa de ser tratado, mas temos algum tempo para planejar como lidar com isso, enquanto que um impedimento com alta urgência e alto impacto precisa de ser tratado imediatamente ou ele vai rapidamente ter um grande impacto negativo sobre o projeto ou programa.

2.3.2.3. Avaliação do impedimento

A avaliação do impedimento é a etapa onde podemos comparar os vários valores de urgência e de impacto para todos os impedimentos que nos permite priorizar a ordem em que vamos abordar os impedimentos.

2.3.2.4 Implementação de ações

A etapa Implementação de ações é onde nós documentamos a ação para resolver o impedimento, em seguida; a ação é implementada.

2.3.2.5 Controlo e revisão

É aqui que são revistos e atualizados todos os impedimentos, para verificar se estes foram resolvidos e os podemos retirar do catálogo, ou para verificar se a urgência ou o impacto do impedimento mudou caso este ainda não esteja resolvido.

3. Sistemas de Suporte à Decisão

Neste capítulo será esclarecido o que é uma Decisão e como estas podem ser classificadas.

Em seguida são abordados os Sistemas de Suporte à Decisão, o conceito, a sua história, taxonomias e arquiteturas.

No final é feita uma conclusão do conteúdo analisado.

3.1 Decisão e processo de decisão

A decisão é um caminho ou uma resolução que se toma acerca de um determinado assunto. Regra geral, a decisão supõe iniciar ou pôr fim a uma situação. Isto é, impõe uma mudança de estado.

Segundo Nascimento e Reginato [37], as decisões podem ser classificadas como:

- Decisões estruturadas;
- Decisões semi-estruturadas;
- Decisões não estruturadas.

Decisões estruturadas são decisões repetitivas, com passos previsíveis, como por exemplo contratação de serviços para manutenção de rotina. Estas normalmente referem-se às camadas de gestão operacional.

Decisões semi-estruturadas dizem respeito à camada hierárquica de gestão média. São decisões de carácter semi-estruturado a alteração de processos de produção da empresa.

Decisões não estruturadas envolvem em sua grande parte situações novas para a organização, ou variáveis sobre as quais se tem pouco ou nenhum controle, exigindo, desse modo, um alto grau de inferências e de experiência, percepção e intuição daqueles por elas responsáveis. Estas são normalmente associadas aos executivos da empresa.

A tomada de decisões passa por quatro estados diferentes, “investigação”, “desenho”, “escolha” e “implementação”. O estado “investigação” refere-se à identificação do problema. A fase de exploração de várias soluções com vista à resolução do problema é o estado de “desenho”. A especificação da solução escolhida é referente ao estado “escolha”. Finalmente é implementada a solução e monitorizada, “implementação”. Estes quatro passos devem seguir a ordem indicada.

3.2 Conceito de Sistema de Suporte à Decisão

Um Sistema de Suporte à Decisão pode ser visto como um sistema que analisa um conjunto de variáveis para que seja possível posicionar-se em relação a uma questão, como por exemplo o GPS.

Para Little [38] um SSD é “um modelo base com um conjunto de procedimentos para processar dados e julgamentos de modo a ajudar o decisor na sua decisão final”. Little [39] argumenta que para que um SSD tenha sucesso precisa de ter as seguintes características:

- Simplicidade;
- Robustez;
- Facilidade de controlo;
- Adaptabilidade;
- Facilidade de comunicação com o utilizador.

Um SSD pode englobar várias áreas de conhecimento como ilustrado na figura 3.



Fig.3 - Áreas de conhecimento envolvidas nos Sistemas de Suporte à Decisão

3.3 Breve História dos Sistemas de Suporte à Decisão

Os Sistemas de Suporte à Decisão (SSD) vêm sendo investigados há mais de 40 anos [27].

A história dos SSD começou por volta dos anos 60, quando surgiu o IBM System 360, e outros sistemas *mainframe* poderosos. Estes primeiros sistemas focavam-se em fornecer relatórios periódicos, mas os sistemas não forneciam suporte interativo para ajudarem os utilizadores a decidirem.

Por volta de 1970 as revistas da área da gestão de negócios começam a publicar artigos relacionados com sistemas de decisão de gestão, sistemas de planeamento estratégico e sistemas de apoio à decisão.

No começo dos anos 80 é considerado que o conceito SSD começou a ser estudado intensamente.

No início dos anos 90, um grande avanço tecnológico ocorreu nos SSD, que passaram de SSD baseados em *mainframes* para SSD baseados em cliente/servidor. Nesse mesmo período começaram a surgir a partir do SSD os conceitos de armazém de dados (*data warehouse*) e processamento analítico *on-line* (OLAP).

Em 1995, os armazéns de dados e Internet começam a ter impactos práticos. SSD baseados em Web e com Web ativa começam a ser viáveis em meados de 1995.

Devido à Internet o desenvolvimento dos SSD tem crescido a um ritmo elevadíssimo, no entanto a sua história ainda é curta estando ainda os conceitos e tecnologias a evoluir.

3.4 Taxonomias

Não existe uma taxonomia universal de SSD. Diferentes autores possuem diferentes propostas de classificação. Segundo Hättenschwiller (citado em [10]) existem três tipos de SSD: os passivos, os ativos e os cooperativos. Um SSD passivo é apenas um sistema que trata informação e auxilia na tomada de decisão, mas não apresenta soluções ou sugestões. Um SSD ativo já permite apresentar soluções ou sugestões; é aqui, segundo Hättenschwiller, que se enquadra o SSD desenvolvido no âmbito desta dissertação, pois este apresenta para o utilizador soluções/ações. Um SSD cooperativo permite ao utilizador modificar, completar ou refinar sugestões de outros colaboradores.

Power [29] por seu lado, considera que existem cinco tipos de SSD: orientados por comunicação, orientados por documentos, orientados por dados, orientados por modelos e orientados por conhecimento. Um SSD orientado por comunicação facilita a partilha de informação entre grupos, está otimizado para trabalho em grupo. Já um SSD orientado por

documentos consiste numa gestão de informação não estruturada numa variedade de formatos eletrónicos. Por seu lado, um SSD orientado por dados é normalmente um sistema de manipulação de dados internos à empresa. Um SSD orientado por modelos destaca o acesso e manipulação de modelos de estatística, financeiros, de simulação e de otimização. Usam informação provida pelos utilizadores para auxiliar na tomada de decisão. Por fim, um SSD orientado por conhecimento provê resoluções de problemas a partir de conhecimentos especializados, armazenados como fatos e resoluções comprovadamente boas. Então, de acordo com os princípios enunciados por Power, o SSD desenvolvido enquadra-se nos SSD orientados por conhecimento, uma vez que, o SSD desenvolvido apresenta soluções/ações provenientes do conhecimento de especialistas.

3.5 Arquiteturas

Segundo Sprage e Carlson (citado em [3]) existem três componentes fundamentais num SSD:

- Base de dados;
- Modelo;
- Interface.

A base de dados é onde é armazenada a informação. Pode ser um repositório remoto, um repositório organizacional tradicional, ou com acesso através da internet. O modelo é o contexto e os critérios para solucionar problemas de decisão. A interface é o meio por onde o utilizador comunica com o sistema.

Segundo Power (citado em [3]) são quatro as componentes fundamentais num SSD:

- Interface com o utilizador;
- Base de dados;
- Modelo e ferramentas analíticas;
- Arquitetura e rede de trabalho de SSD.

Existem vários meios de classificação de um SSD. Nem sempre um SSD se enquadra numa das seguintes categorias, podendo ser um misto de uma ou mais.

Holsapple e Whinston (citado em [3]) avaliam os SSD dentro de seis *frameworks*:

- SSD orientado ao contexto;

- SSD orientado à base de dados;
- SSD orientado à folha de cálculo;
- SSD orientado à solução;
- SSD orientado às regras de negócio;
- SSD complexos.

A classificação mais comum para um SSD é um SSD complexo, pois é um sistema híbrido, que engloba duas ou mais estruturas.

Como em qualquer outro sistema, para estruturar um SSD é necessário ter em conta as pessoas, a tecnologia e as técnicas de desenvolvimento. Em relação à tecnologia é necessário ter em atenção os seguintes pontos:

- Interface com o utilizador, pois é com base no que visualiza que o utilizador toma medidas;
- Técnicas de desenvolvimento, de forma a facilitar o desenvolvimento do SSD;
- Ferramentas de desenvolvimento, bibliotecas, as linguagem de programação e módulos, de modo a que o SSD seja robusto e modular.

3.6 Conclusão

Os Sistemas de Suporte à Decisão são utilizados para apoiar o utilizador na tomada de decisão, e não substituir o seu juízo final, apresentando dados ou possíveis soluções relativamente a um determinado problema. Após uma análise das taxonomias, segundo Power, constatou-se que o modelo SSD orientados por conhecimento seria o mais apropriado para o projeto em questão, uma vez que o SSD desenvolvido terá que apresentar soluções/ações provenientes do conhecimento de especialistas. Segundo a classificação de Hättenschwiller um SSD ativo é o necessário para o projeto, pois o SSD desenvolvido terá que apresentar, explicitamente para o utilizador, soluções/ações.

Na pesquisa efetuada não se encontrou nenhum sistema de suporte à decisão especificamente para o tratamento de riscos e impedimentos no contexto da gestão de projeto (objetivo deste trabalho).

4. Concepção e Implementação

4.1 Estratégia e Funcionalidades

O módulo desenvolvido tem que ser integrado na vertente *mobile* do SCRAIM (plataforma Web), o AMOB. De acordo com os problemas inseridos pelos utilizadores na plataforma Web, impedimentos ou riscos, associados a um determinado projeto em que o utilizador está inserido, o módulo deverá retornar possíveis soluções para impedimentos e possíveis ações de mitigação para riscos, sendo estas baseadas num catálogo fornecido por um especialista. O módulo tem que permitir também a introdução de soluções ou ações de mitigação por parte do utilizador, para uso futuro.

Um utilizador do SCRAIM, ao aceder aos impedimentos ou riscos associados ao projeto, terá a possibilidade de consultar as propostas de solução dos impedimentos e as propostas de ação de mitigação para os riscos.

Como as soluções ou ações propostas pelo sistema têm que ser de acordo com os problemas, impedimentos ou riscos, inseridos pelo utilizador, a estratégia passará por reconhecer palavras-chave no problema e de acordo com essas palavras-chave retornar possíveis soluções/ações correspondentes no catálogo existente. De modo a que o algoritmo seja mais completo e abrangente é necessário também que este reconheça variações das palavras-chave para mais possibilidades de “*match*”. A cada solução/ação presente no catálogo estará associado um conjunto de palavras-chave. Sendo que um & (and) de palavras significa que essas palavras têm obrigatoriamente de estar presentes no texto do problema e um | (or) entre palavras ou conjunto de palavras significa que basta estar aquela palavra ou aquele conjunto de palavras presentes na descrição do problema para este ser opção. Seguem-se alguns exemplos ilustrativos. A figura 4 apresenta também o fluxo de funcionamento concebido para o sistema.

Exemplo 1:

Risco introduzido pelo utilizador:

“Missing skills on the team”

Possível solução na base de dados (catálogo) com as palavras-chave associadas:

Ação: “Schedule training for team members”

Palavras-chave: “missing&skills&team”

Ou seja, como o risco tem na sua descrição as palavras “missing”, “skills” e “team” a ação na base de dados que será apresentada será “Schedule training for team members”, sendo que, se as três palavras não estiverem na descrição do risco esta solução não será apresentada, uma vez que resulta de uma conjunção das três palavras.

Exemplo 2:

Impedimento introduzido pelo utilizador:

“Absent product owner holding up key decisions”

Possíveis solução na base de dados com as palavras-chave associadas:

Solução: “ Contact product owner”

Palavras-chave: “absent&owner|problem&owner”

Solução: “ Scale to upper management”

Palavras-chave: “absent&product&owner&holding&decisions”

Ou seja, a primeira solução será apresentada quando as palavras “absent” e “owner” aparecerem na descrição do impedimento **ou** aparecerem as palavras “problem” e “owner”, uma vez que é um “|” (*or*) do conjunto das palavras.

Para este caso serão apresentadas as duas soluções pois existem na base de dados duas soluções contendo as palavras-chave detetadas na descrição do impedimento.

Se o sistema não conseguir fazer nenhum tipo de “*match*”, apresenta uma solução/ação *standard*, como por exemplo “Schedule a meeting”.

É também interessante o sistema priorizar as soluções/ações, principalmente as introduzidas pelos utilizador.

As soluções/ações introduzidas pelos utilizadores têm também que ser acompanhadas pelas respetivas palavras-chave. Pois, olhando a um exemplo extremo, caso um utilizador tivesse um problema em que um dos seus colaboradores tivesse falecido aqui certamente a palavra-chave seria faleceu ou um conjunto de esta com outras; assim, é dado, ao utilizador, a liberdade de ser ele a escolher as palavras-chave associadas a determinado problema.

Estas soluções serão guardadas na base de dados no mesmo formato que as soluções já presentes no catálogo.

A figura seguinte resume o funcionamento geral.

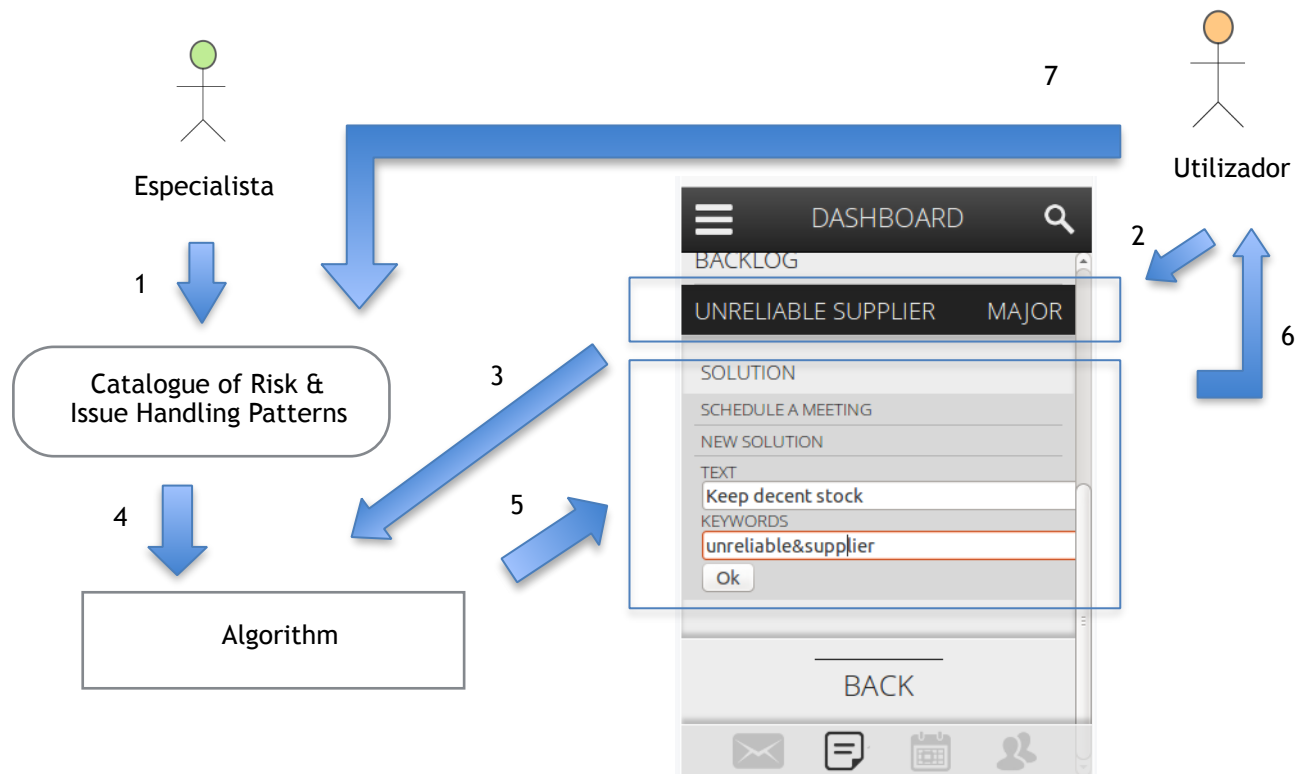


Fig.4 - Funcionamento geral

1. Através do conhecimento de um especialista é definido um catálogo base;
2. O utilizador descreve o problema;
3. O algoritmo analisa a descrição do problema;
4. O algoritmo verifica a correspondência no catálogo;
5. O algoritmo retorna as possíveis soluções/ações para resolução do problema;
6. O utilizador toma conhecimento das possíveis soluções/ações para resolução do problema;
7. Se o utilizador assim entender pode contribuir para o catálogo.

Há dois atores principais envolvidos no sistema:

- Especialista é aquele que revê/constrói o catálogo base do sistema, com o conjunto solução - palavras-chave.
- Utilizador é aquele que interage com o sistema no dia a dia.

A figura 5 resume as principais funcionalidades a disponibilizar a estes atores pelo módulo SSD.

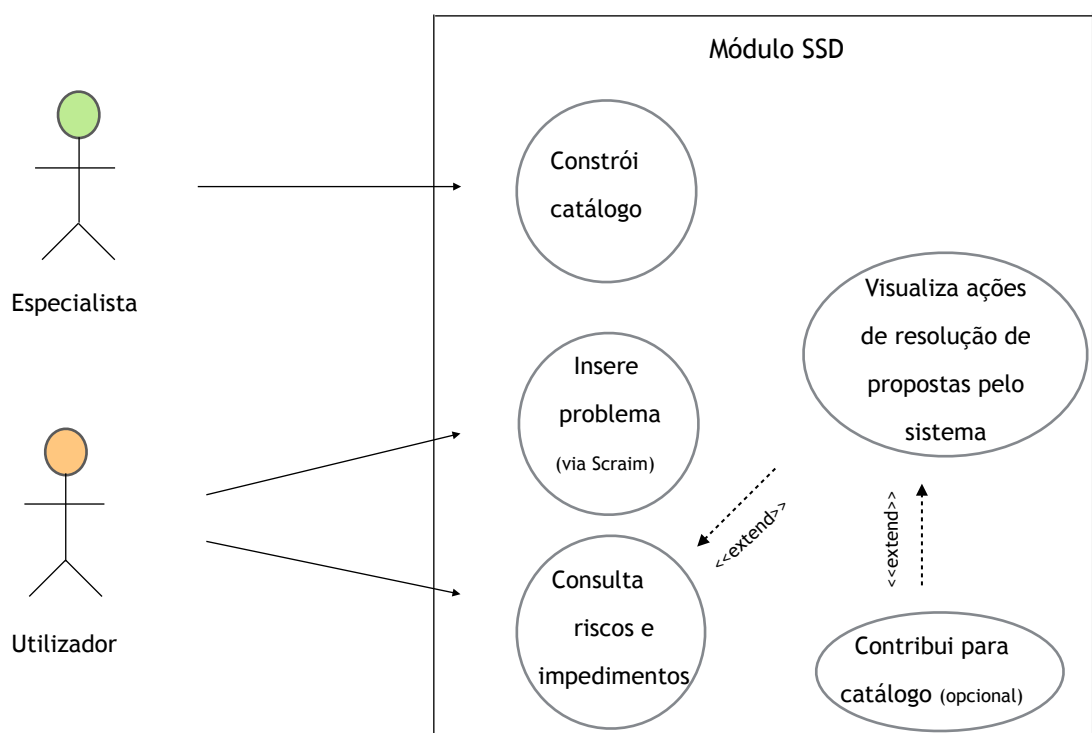


Fig.5 - Diagrama de casos de uso UML referente ao SSD a desenvolver

4.2 Arquitetura e Tecnologias

4.2.1 Arquitetura

O sistema desenvolvido utiliza a mesma estrutura do SCRAIM para o armazenamento de dados. A Figura 6 representa a arquitetura geral do sistema.

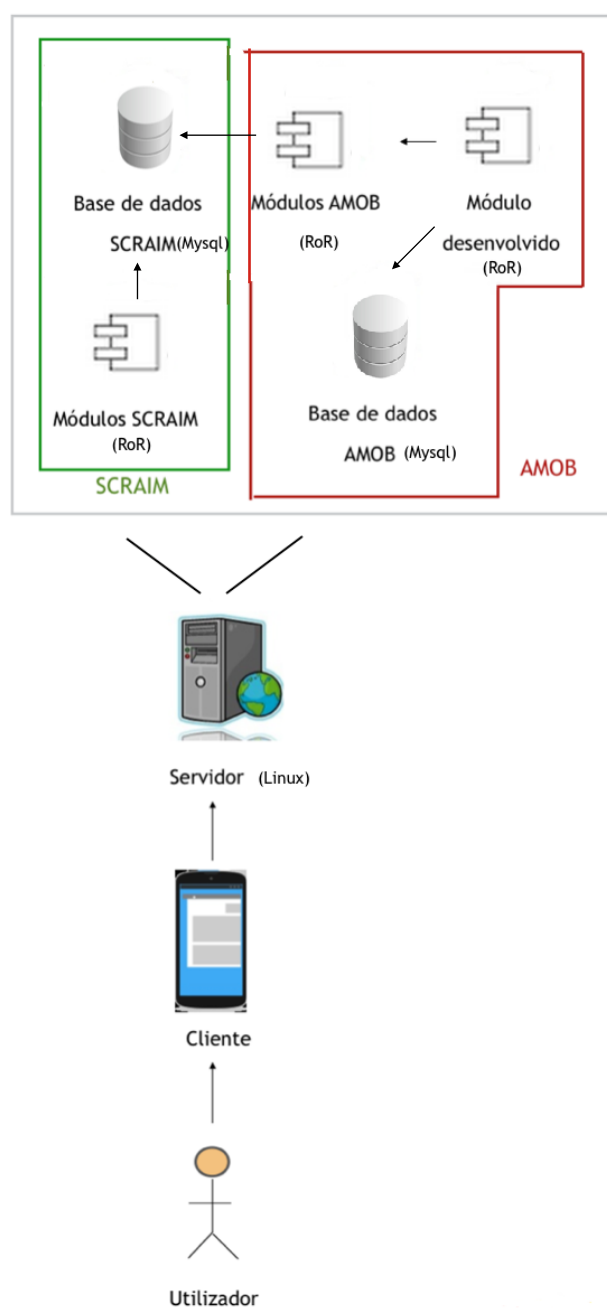


Fig.6 - Arquitetura geral da solução desenvolvida

4.2.2 Tecnologias

Uma vez que o módulo a desenvolver é para integrar o sistema já existente na StrongStep, as tecnologias estão limitadas às já utilizadas no SCRAIM/AMOB. As tecnologias utilizadas são o Ruby on Rails e JavaScript. É necessário considerar que parte do tempo dispensado na dissertação foi na aprendizagem destas tecnologias, visto serem completamente novas para mim, tendo sido este, como já referido, um dos pontos de motivação para a realização deste trabalho.

4.2.2.1 Ruby on Rails

O criador de Ruby on Rails (RoR) foi David Heinemeier Hansson, sendo que o RoR foi lançado ao público pela primeira vez em 2003.

Desde o seu lançamento, Ruby on Rails (RoR) tornou-se rapidamente num dos *frameworks* mais poderosos e populares para a construção de aplicações web. Desde grandes empresas a pequenas *startups* têm utilizado o RoR. É um *framework* livre, escrito na linguagem de programação Ruby. As aplicações são desenvolvidas com base no padrão de arquitetura MVC (*Model-View-Controller* Fig.7).

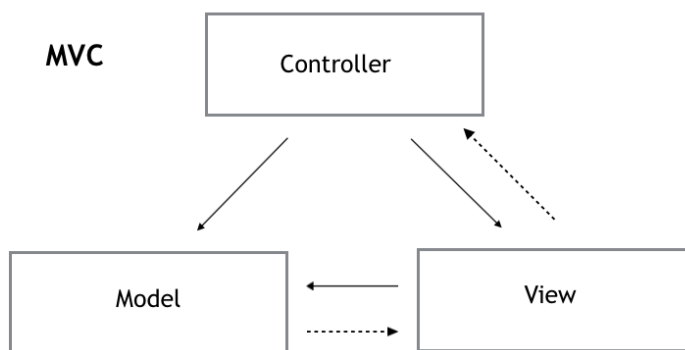


Fig.7 - Arquitetura MVC

As setas sólidas indicam associação direta e as linhas a tracejado indicam associação indireta (Figura 8).

Um **controlador** (*controller*) pode enviar comandos para a vista (*view*) associada para alterar a apresentação da vista do modelo. O controlador também pode enviar comandos para o modelo (*model*) para atualizar o estado do modelo. É prática corrente dar nomes aos controladores de forma que eles representem o modelo que eles controlam seguido por

_controller.rb (Ex: login_controller.rb). Se esta regra não for seguida, o mapeamento automático de URL não ocorrerá.

Um **modelo** (*model*) permite o acesso a dados presentes em diferentes fontes de dados, usualmente bases de dados, bem como o seu armazenamento e manipulação.

A **vista** (*view*) define a forma como a informação é apresentada ao utilizador, solicitando a informação ao modelo.

4.2.2.2 JavaScript

JavaScript (JS) foi originalmente desenvolvido por Brendan Eich, em 1995.

O JavaScript permite que a experiência de utilização de uma página Web seja muito mais eficaz, simples e agradável. Os *scripts* desenvolvidos em JavaScript são muito populares e amplamente integrados em páginas web devido à facilidade de interação com o Document Object Model (DOM) da página. É uma linguagem difícil de utilizar pois não tem uma convenção que defina todas as funcionalidades para todos os *browsers*.

4.3 Estrutura da Base de Dados

De forma a assegurar o armazenamento da informação foi criada uma base de dados, seguindo a convenção Rails. O modelo da base de dados foi desenhado com restrições de maneira a garantir a integridade dos dados. Na tabela seguinte, são apresentados os campos associados à tabela da base dados do AMOB, que é a tabela das soluções/ações e das palavras-chave. O campo solução/ação tem que ser único na base de dados, para assim não serem duplicadas soluções/ações. O campo keywords também tem que estar presente, e para além disso tem que ter um certo formato; para isso foi criada uma expressão regular (`/\A([a-zA-Z]{3,})+(\&|\|[a-zA-Z]{3,})*\z/i`), que impede o utilizador de introduzir palavras com menos de três caracteres bem como símbolos estranhos ao algoritmo.

Tabela 2 - Tabela Soluções/Ações (AMOB)

Tabela Solutions				
Campos	Tipo	Null	Default	Extra
id	integer	NO	NULL	<i>auto_increment</i>
keywords	string	NO		
count	integer	NO	0	
text	string	NO		
type	string	YES		
user_id	integer	YES	NULL	
created_at	datetime	NO		
updated_at	datetime	NO		

Foram também utilizadas algumas tabelas da base de dados do SCRAIM, a tabela dos Issues, que é a tabela que contém informação relativa aos problemas dos utilizadores (impedimentos/risco). E a tabela Users que é a tabela que contém a informação relativa aos utilizadores. (Tabelas 3 e 4)

Tabela 3 - Tabela Issues (SCRAIM)

Tabela “Issues”				
Campos	Tipo	Null	Default	Extra
id	integer	NO	NULL	<i>auto_increment</i>
tracker_id	integer	NO	NULL	
project_id	integer	NO	NULL	
subject	varchar	NO		
description	string	YES	NULL	
due_date	date	YES	NULL	
category_id	integer	YES	NULL	
status_id	integer	NO	NULL	
assigned_to_id	integer	YES	NULL	
priority_id	integer	NO	NULL	
fixed_version_id	integer	YES	NULL	

author_id	integer	NO	NULL	
lock_version	integer	NO	0	
created_on	datetime	YES	NULL	
updated_on	datetime	YES	NULL	
start_date	date	YES	NULL	
done_ratio	integer	NO	0	
parent_id	integer	YES	NULL	
root_id	integer	YES	NULL	
lft	integer	YES	NULL	
rgt	integer	YES	NULL	
is_private	tinyint	NO	0	
closed_on	datetime	YES	NULL	
aims_iteration_id	integer	YES	NULL	
issue_type	varchar	YES	NULL	
priority_order	integer	YES	0	
important	tinyint	YES	0	
actual_end_date	date	YES	NULL	
estimated_end_date	date	YES	NULL	
dimension_id	integer	YES	NULL	
external_work_product	tinyint	YES	NULL	
aims_project_wbs_id	integer	YES	NULL	
external_id	varchar	YES	NULL	
original_tracker	varchar	YES	NULL	
estimated_hours	float	YES	NULL	
remaining_hours	float	YES	NULL	

Tabela 4 - Tabela Users (SCRAIM)

Tabela Users				
Campos	Tipo	Null	Default	Extra
id	integer	NO	NULL	auto_increment
Login	varchar	NO		
hashed_password	varchar	NO		
firstname	varchar	NO		
lastname	varchar	NO		
mail	varchar	NO		
admin	tinyint	NO	0	
status	integer	NO	1	
Last_login_on	datetime	YES	NULL	
language	varchar	YES		
auth_source_id	integer	YES	NULL	
created_on	datetime	YES	NULL	
updated_on	datetime	YES	NULL	
type	varchar	YES	NULL	
identity_url	varchar	YES	NULL	
mail_notification	varchar	NO		
salt	varchar	YES	NULL	
menu_favourites	varchar	YES	Overview, Dashboard, Planning	
must_change_password	tinyint	NO	0	

4.4 Ficheiros criados e modificados

Neste sub-capítulo serão identificados quais os ficheiros criados e modificados no AMOB, para uma melhor manutenção e gestão do módulo desenvolvido.

Ficheiros modificados:**Gemfile**

É uma lista de todas as *gems* que desejamos incluir no projeto. Aqui foram adicionadas *gems* como a “fast-stemmer”.

app/helpers/application_helper.rb

Contém classes auxiliares e *helpers*, tais como a “Hash_DB”.

app/models/risk.rb

É onde são acedidos os riscos dos projetos.

app/models/user.rb

É onde se encontra a informação acerca dos utilizadores.

app/views/project_impediments_risks/_impediments.html.erb

É onde se encontra o *partial* para o *layout* dos impedimentos.

app/views/project_impediments_risks/_risk.html.erb

É onde se encontra o *partial* para o *layout* dos riscos.

app/views/project_impediments_risks/display.html.erb

É onde se encontram algumas funções para o bom funcionamento do *layout*.

config/routes.rb

Contém arquivos de configuração para o ambiente Rails, neste caso o mapa de roteamento.

Ficheiros criados:**app/controllers/solutions_controller.rb**

É o controlador das soluções/ações.

app/models/solution.rb

É o modelo das soluções/ações, é por exemplo onde se encontram as restrições que garantem a integridade dos dados.

db/migrate/[timestamp]_create_users.rb**db/migrate/[timestamp]_create_solutions.rb**

db/migrate/[timestamp]_add_count_to_solutions.rb

db/migrate/[timestamp]_add_user_id_to_solutions.rb

São ficheiros que contêm migrações para o esquema da base de dados.

4.5 Algoritmo de Recomendação

4.5.1 Base do algoritmo desenvolvido

Neste sub-capítulo será apresentado a base do algoritmo, em pseudocódigo. A base do algoritmo é o mecanismo de “*matching*”, em que é analisada a descrição do problema e comparada com o conjunto solução/palavras-chave que se encontra na base de dados. Foi também criado um método auxiliar que transforma os dados da base de dados antes de estes serem analisados pela base do algoritmo. O método criado, o “Hash_DB”, transforma o que está na base de dados para uma hash de vetores de modo a que o algoritmo base funcione como pretendido.

Numa **primeira fase**, o algoritmo vai “buscar” à base de dados todos os conjuntos solução/ação - palavras-chave, e vai separar o campo “keywords”, primeiro pelo | (or), e em seguida pelo & (and). É obtido um vetor com todas as “keywords” correspondentes a cada solução/ação, onde, dentro desse vetor se encontra outro vetor com as palavras ou conjunto de palavras separadas pelo | (or) onde, dentro deste vetor, se encontra outro vetor com as palavras separadas pelo & (and). Pois pode ser um | (or) com um conjunto de palavras (ex: cliente&dados|projeto&atraso), sendo necessário contar o número de palavras “dentro” de cada | (or).

Convenções:

f[j] -> j é a posição no vetor f

f[i][j] -> significa um vetor dentro de outro vetor e j e i são as respetivas posições

Para **cada** problema e para **todos** os conjuntos solução - palavras-chave:

Pseudo-código:

Entradas

S -> conjunto de pares soluções/palavras-chave

Saídas

filters :array (triplo)

ÍNICIO

VARIÁVEIS AUXILIARES

ret,g,f :array

PARA i=0 ATÉ tamanho total de S FAÇA

 g <- Substrings de S separadas pelo |

 PARA j=0 ATÉ tamanho total de g FAÇA

 f[j] = Palavras em g[j] separadas pelo &

 FIM

 filters <- f

FIM

Na **segunda fase** é apresentado todo o mecanismo de “*match*”.

Num primeiro momento desta fase, é verificado para cada posição do filters (“PARA i=0 ATÉ tamanho total de filters FAÇA”) qual é o seu tamanho, para averiguar se as palavras-chave foram ou não separadas pelo | (or), ou seja, se o tamanho for igual a um quer dizer que as palavras não foram separadas pelo | (or), então, é uma palavra ou é um conjunto de palavras separadas por & (and). De seguida é verificado quantas palavras são no total (uma palavra ou um conjunto de palavras), se for um conjunto de palavras é porque é um & (and) de palavras, assim, todas as palavras têm que fazer “*match*” (é usado o método match para procurar uma palavra numa determinada *string*), se for apenas uma palavra apenas é necessário essa palavra fazer match, e caso isso se verifique é adicionado a um vetor a solução para quais as palavras-chave fizeram “*match*”.

Continuação:

Pseudo-código:

VARIÁVEIS

continue=true, continue_processing=true :boolean

Nota: as variáveis continue e continue_processing são para controlo de ciclos
elem :string

PARA i=0 ATÉ tamanho total de filters FAÇA

VARIÁVEIS

count=0, size=0, comprimento :inteiro

SE o tamanho do filters na posição i for igual a 1 ENTÃO

size=tamanho de filters[i][0]

PARA k=0 ATÉ size-1 ENTÃO

elem=filters[i][0][k]

SE elem não estiver no problema ENTÃO

continue_processing=false

QUEBRAR CICLO

SENÃO

count=count+1

SE k for igual ao size-1 ENTÃO

SE count for igual ao size ENTÃO

continue_processing=true

colocar em ret a solução correspondente ao “match”

QUEBRAR CICLO

FIM

FIM

FIM

FIM

Em seguida, é quando existe pelo menos um | (or), inicialmente é verificado se entre | (or's) é apenas uma palavra, e caso essa palavra faça “match” é adicionado a um vetor a solução correspondente. Caso entre | (or's) seja mais que uma palavra, ou seja, é um & (and) de palavras, então, todas essas palavras tem que fazer “match” e caso isso se verifique é adicionada a respetiva solução.

Continuação:

Pseudo-código:

SENÃO

comprimento= tamanho do filters[i]

PARA j=0 ATÉ tamanho total do filters[i] FAÇA

SE continue é igual a false ENTÃO

QUEBRAR CICLO

SENÃO

SE o tamanho de filters[i][j] é igual a 1 ENTÃO

elem = filters[i][j][0]

SE elem estiver no problema ENTÃO

continue_processing = true

colocar em ret a solução correspondente ao “match”

SENÃO

count=count+1

SE count for igual ao comprimento ENTÃO

continue_processing=false

QUEBRAR CICLO

FIM

FIM

SENÃO

size=tamanho do filters[i][j]

PARA k=0 ATÉ ao tamanho total do filters[i][j] FAÇA

elem=filters[i][j][k]

SE elem não estiver no problema ENTÃO

continue_processing=false

QUEBRAR CICLO

SENÃO

count=count+1

SE count for igual ao size

continue_processing=true

colocar em ret a solução correspondente ao “match”

continue=false

QUEBRAR CICLO

FIM

FIM

FIM

FIM

FIM

```

FIM
FIM
SE continue_processing for igual a false ENTÃO
    continue_processing=true
    SALTAR IMEDIATAMENTE PARA A PRÓXIMA ITERAÇÃO DO CICLO
FIM
FIM
RETORNAR ret
FIM

```

4.5.2 Redução das palavras à sua raiz

Neste sub-capítulo será abordado o que foi feito para tornar o algoritmo mais abrangente.

Para tornar o algoritmo mais abrangente recorreu-se a uma *gem* a “fast-stemmer”, que é uma *gem* que contém uma biblioteca (em inglês) muito vasta de palavras, na sua forma raiz e reduz as mesmas para esta forma. A “fast-stemmer” é baseada no algoritmo de Porter “*The Porter Stemming Algorithm*” [30]. A raiz da palavra é o constituinte da palavra que contém significado lexical mas não inclui afixos derivacionais ou flexionais. No entanto a “fast-stemmer” teve que ser adaptada para se obter os resultados pretendidos, pois a “fast-stemmer” apenas transforma uma palavra para a sua raiz de cada vez, e para que fosse possível transformar frases inteiras foi criado o método auxiliar “stem_phrase” que funciona em conjunto com a “fast-stemmer”.

4.5.3 Ordenação das soluções

O módulo desenvolvido também ordena as soluções/ações de acordo com o valor da variável count (Tabela 2); este valor é definido pelo especialista com base no seu julgamento em relação a determinada solução/ação. Por norma quantas mais palavras-chave mais pontuação terá a solução/ação, mas este aspeto é deixado ao critério do especialista.

4.5.4 Sistema de tradução

Foi também implementado um sistema de tradução, onde a descrição do problema em português é analisada e traduzida para inglês, sendo depois analisada pelo módulo já em inglês.

Para isto recorreu-se a uma *gem* a “ms-translator” que é uma API da Microsoft. Foi também necessário desenvolver um método auxiliar o “translate_phrase” que é um método de configuração da “ms-translator”; estes funcionam em conjunto para se obter o resultado pretendido. No entanto esta funcionalidade apesar de estar implementada ainda não está a ser utilizada, pois fará parte de uma vertente *premium* do SCRAIM/AMOB.

4.6 Catálogo de Soluções

Nesta dissertação foi também proposto um catálogo base (Tabela 5) para o módulo implementado, sendo previamente revisto e aceite pelo especialista da StrongStep.

Tabela 5 - Catálogo de Soluções

Catálogo		
Palavras-chave	Solução Proposta	Exemplo
Management & commitment	Schedule meeting with top management to present project report/status	Lack of top management commitment to the project
Requirements	Schedule meeting to review the requirements	Misunderstanding of the requirements
management & change	Review planning and releases	Not managing change properly
absent & owner problem & owner	Contact product owner	Absent product owner holding up key decisions
absent & product & owner & holding & decisions	Scale to upper management	Absent product owner holding up key decisions
missing & skills & team	Schedule training for team members	Missing skills on the team
unreliable & supplier shortfalls & components	Keep decent stock	Unreliable Supplier

Catálogo		
failure & user & commitment	Gather evidence of user commitment acceptance	Failure to gain user commitment
absent & user problem & user	Schedule meeting with user	Absent User
management & skill & lack	Plan training sessions	Lack of effective project management skill
Scope objective	Have approved and versioned project scope/ objectives document	Unclear objectives
unrealistic & time unrealistic & deadlines poor & planning insufficient & resources	Create realistic and detailed agenda.	Unrealistic time and cost estimates
developing & wrong	Schedule weekly meetings between team members	Developing the wrong software functions
gold & plating real-time & performance & shortfalls	Prototyping	Gold Plating
personnel & shortfall	Staffing with top talent	Personnel shortfall
shortfall & task	Award-fee contracts	Shortfalls in externally performed tasks
straining & capabilities	Make technical analysis	Straining Science Capabilities
inadequate & features & security	Create realistic and detailed plan of security	Inadequate security features being built into the system
software & outdated	Analyzes the cost-benefit of buying or renting new software	Software becoming outdated
conflicts & team conflicts & department	Schedule activities between members	Personal conflicts between team members

5. Apresentação da Solução Desenvolvida

Neste capítulo serão apresentados os resultados finais, obtidos durante a realização desta dissertação. Primeiro serão apresentados os resultados obtidos para os riscos e em seguida para os impedimentos. É importante voltar a referir que os impedimentos e riscos são adicionados no SCRAIM (plataforma Web). Nesta fase, o catálogo é preenchido por acesso direto à base de dados, através da consola.

5.1 Consulta de riscos e ações de mitigação

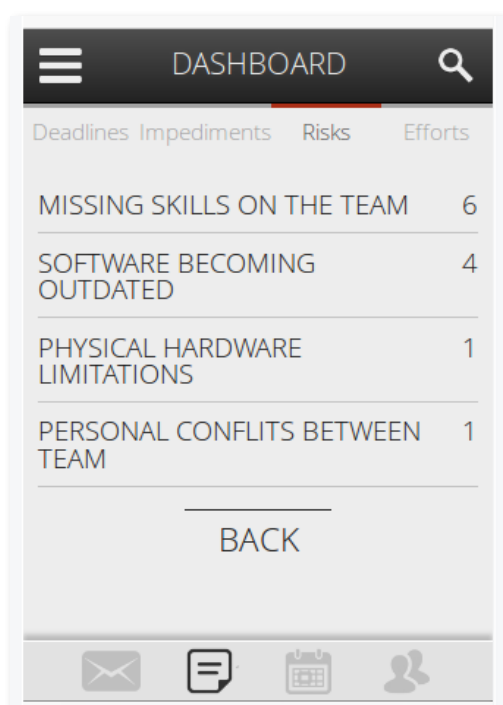


Fig.8 - Vista detalhada de todos os riscos

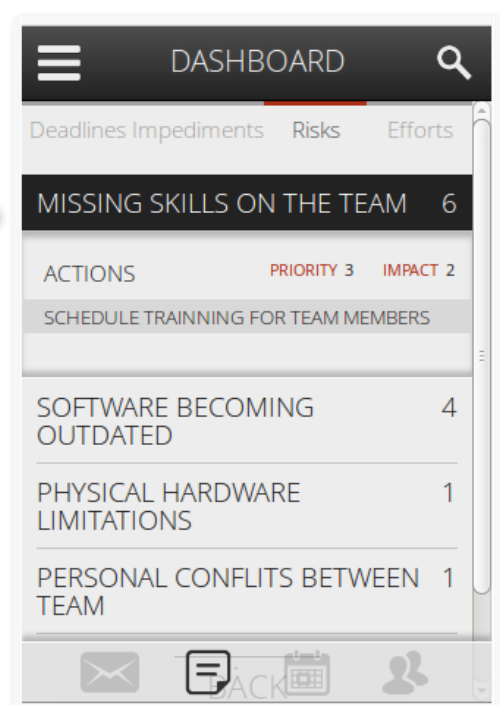


Fig.9 - Ação para o risco selecionado

Neste exemplo o módulo encontra na base de dados uma ação de mitigação para o conjunto de palavras que formam o risco (Ver figura 9).

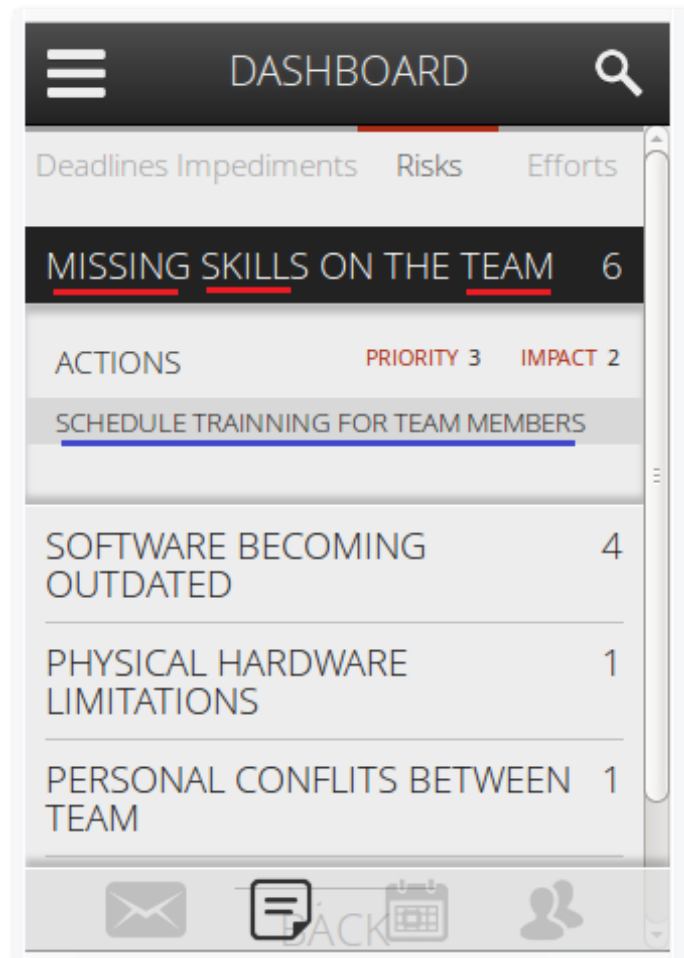


Fig.10 - Palavras-chave mais Ação para o risco
“Missing skills on the team”

O conjunto Ação/Palavras-chave que se encontra na base de dados para o risco “Missing skills on the team” é o seguinte:

Solução: “Schedule training for team members”

Palavras-chave: “miss&skill&team”

Ou seja, como referido anteriormente, apesar de na descrição do risco as palavras-chave encontradas serem “missing”, “skills” e “team”, estas são reduzidas à sua forma raiz e analisadas como “miss”, “skill” e “team”.

5.2 Consulta de impedimentos e ações sugeridas

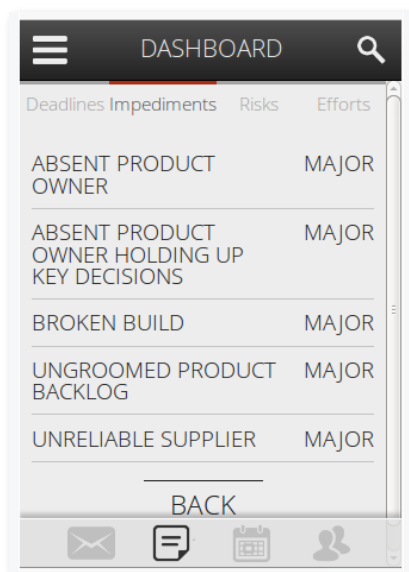


Fig. 11 - Vista detalhada de todos os impedimentos

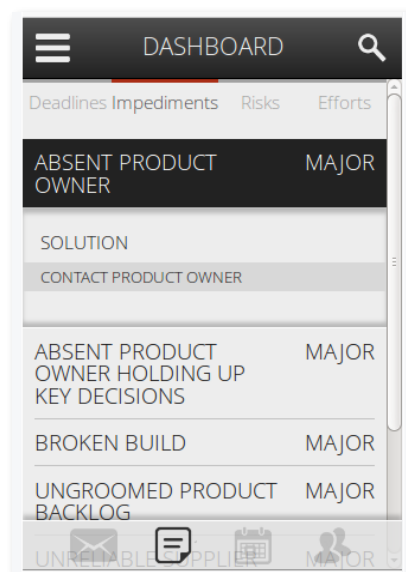


Fig. 12 - Solução para o impedimento selecionado

No exemplo seguinte o módulo encontra na base de dados duas soluções para o conjunto de palavras que formam o impedimento.

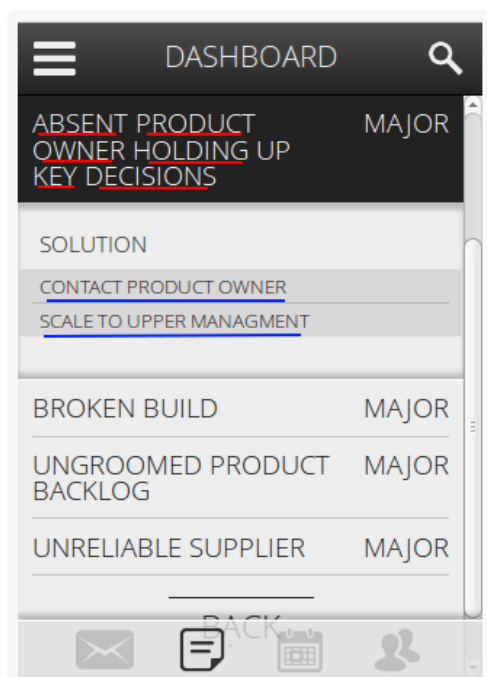


Fig. 13 - Palavras-chave mais soluções para o impedimento “Absent product owner holding up key decisions”

O conjunto Solução/palavras-chave que se encontra na base de dados para o impedimento “Absent product owner holding up key decisions” é o seguinte:

Solução: “ Contact product owner”

Palavras-chave: “absent&owner|problem&owner”

Solução: “ Scale to upper management”

Palavras-chave: “absent&product&owner&hold&decis”

Neste exemplo foram retornadas duas soluções, priorizadas. (A variável contar é o que prioriza as soluções quanto mais alta mais prioridade tem essa Solução - Ver tabela 1).

No exemplo seguinte não houve “match” em relação ao impedimento introduzido pelo utilizador (“Unreliable Supplier”), então é retornada uma solução *standard* (“Schedule a meeting”). Neste caso o utilizador optou por colocar a sua própria solução com as palavras-chave que ele achou adequadas ao impedimento. Na base de dados é guardada a solução com as palavras-chave “unreliable” e “supplier” na sua forma raiz sendo que, numa próxima utilização, caso apareçam estas palavras-chave, esta solução terá prioridade elevada sobre as outras (Ver Figura 14).

Fig. 14 - Solução introduzida pelo utilizador

6. Conclusões e Trabalho Futuro

6.1 Conclusão

O módulo desenvolvido, pretendia dotar o AMOB de um sistema que ajudasse o utilizador a tomar decisões para os seus problemas (impedimentos/riscos) retornando possíveis soluções/ações, tendo como base o conhecimento de um especialista.

Numa primeira fase, foi necessário entender e aprender as tecnologias Ruby on Rails e JavaScript, sendo que a sua aprendizagem ainda levou algum tempo, pois eram tecnologias completamente novas para mim; no entanto a sua aprendizagem mostrou-se desafiadora e muito interessante, visto também serem tecnologias muito utilizadas hoje em dia. Numa segunda fase foi necessário entender a estrutura do SCRAIM e AMOB, para saber o que poderia ser reutilizado e o que teria de ser modificado e criado. Num terceiro momento (concepção), foi necessário “desenhar” o módulo, como este iria funcionar, e como este iria ser implementado, sendo que, optou-se por algo versátil e de crescimento contínuo, devido à possibilidade de o utilizador contribuir para a base de dados. De seguida, procedeu-se à implementação do módulo; nesta fase constatou-se o aparecimento de muitos *bugs*, sendo que estes foram sempre resolvidos com sucesso. A última fase foi a sua integração no *layout* do AMOB, recorrendo sobretudo ao JavaScript.

Como contribuição foi também proposto um catálogo base de ações-padrão para os riscos e impedimentos.

Em suma, como pretendido, a elaboração desta dissertação permitiu acrescentar valor ao produto desenvolvido e comercializado na StrongStep, sendo que o módulo desenvolvido, tendo como base um bom catálogo, apresenta bons resultados e é fiável.

6.2 Trabalho Futuro

Após a realização desta dissertação, ficam recomendações e alterações que podem ser feitas futuramente.

Por forma a reduzir algum possível incómodo por parte do utilizador na inserção de novas soluções/ações e palavras-chave, seria interessante que o módulo aquando da inserção da solução/ação, automaticamente identificasse as palavras na descrição do problema e através de um *pop-up* pedisse ao utilizador para seleccionar aquelas que ele consideraria chave.

De forma a melhorar o catálogo, seria interessante o utilizador poder dar *feedback*, através de uma *flag*, acerca de determinada solução/ação, para assim se saber se determinada solução/ação tinha sido eficaz e com base nisso melhorar o catálogo.

Referências

1. Jay Heizer; Render Barry. "Operations Management": International Edition, 7ª Edição 2004, ISBN 131209744
2. Victor Lobo, "Sistemas de Apoio à Decisão", EN/ISEGI, 2009
3. Wikipédia. (Jun. 2014) Sistemas de Suporte à decisão Disponível em: pt.wikipedia.org/wiki/Sistema_de_suporte_à_decisão
4. Luciana Miyoko Massukado, "Sistemas de Apoio à Decisão: Avaliação de Cenários de Gestão Integrada de Resíduos Sólidos Urbanos Domiciliares, Universidade Federal de São Carlos, 2004
5. Carlos Alberto Ferreira Bispo; EdsonWalmir Cazarini, "A Nova Geração de Sistemas de Apoio à Decisão" , 1998
6. José Eduardo Malta de Sá Brandão;Joni da Silva Fraga, "Gestão de Riscos", (IPEA) Brasília DF; Departamento de Automação e Sistemas Universidade Federal de Santa Catarina Florianópolis SC
7. Diogo Manuel Mendes Oliveira Santos, "Metodologia de Melhoria Contínua na Gestão de Projetos", (FEUP) Faculdade de Engenharia da Universidade do Porto (Jul. 2013)
8. Wikipédia, (Jun. 2014) Gerência de Projetos Disponível em: pt.wikipedia.org/wiki/Gerência_de_projetos
9. S. Forsman. "OLAP Council White Paper". OLAP Council, 1997
10. Pedro Antunes Silva, "Web-based Management and Decision Support System", (FEUP) Faculdade de Engenharia da Universidade do Porto (Jun 2010)
11. Artan, (Jun.2014) Gestão de riscos Disponível em: artan.com.br/tecnologia/a-gestao-dos-riscos-de-roubo-e-furto-em-empresas
12. Fabio Tomio, (Jun.2014) Ruby on Rails Disponível em: fabiotomio.com.br/blog/2008/12/21/iniciando-em-ruby-on-rails/
13. Funpar, (Jun. 2014) Arquitetura de Software Disponível em: funpar.ufpr.br:8080/rup/process/workflow/ana_desi/co_swarch.htm
14. Wikipédia. (Jun. 2014) JavaScript Disponível em: pt.wikipedia.org/wiki/JavaScript
15. ISO (International Organization for Standardization). ISO 31000 - Risk Management - Principles and Guidelines. 34, Suíça.

16. Valério Antonio Pamplona Salomon “ Desempenho da Modelagem do Auxílio à Decisão por Múltiplos Critérios na Análise do Planejamento e Controle da Produção”, Universidade de São Paulo, 2004
17. Wikipédia. (Jun. 2014) Ruby on Rails Disponível em: pt.wikipedia.org/wiki/Ruby_on_Rails
18. Eco, Umberto, “ Conjunto de Conhecimentos em Gerenciamento de Projetos”, 3. ed. Guia PMBOK
19. Ciberdúvidas da Língua Portuguesa (Jun.2014) Raiz Disponível em: ciberduvidas.com/pergunta.php?id=17358
20. Ruby on Rails Tutorial (Jun. 2014) Disponível em: railstutorial.org/book
21. PMI, Gestão de projetos (Jun.2014) Disponível em: brasil.pmi.org/brazil/AboutUS/WhatIsProjectManagement.aspx
22. Conceito Blog (Jun. 2014), Decisão Disponível em: conceito.de/decisao
23. Artia Blog (Jun. 2014) Disponível em: artia.com/blog/
24. Wikipédia. (Jun. 2014) MVC Disponível em: pt.wikipedia.org/wiki/MVC
25. Diane McGall (Jun, 2014), Issues Management Prevent Crises Available: dianemcgall.wordpress.com/2013/01/14/issues-management-prevents-crises/
26. Significados (Jun. 2014) Disponível em: significados.com.br/projeto/
27. A Brief History of Decision Support Systems (Jun. 2014) Disponível em: dssresources.com/history/dsshhistory.html
28. P. Haettenschwiler, "Neues anwenderfreundliches Konzept der Entscheidungsunterst," Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft, pp.189-208, 1999.
29. Daniel J. Power, "Specifying an Expanded Framework for Classifying and Describing Decision Support Systems," Communications of the Association for Information Systems, vol. 13, no. 13, pp. 158-166, Fevereiro 2004.
30. The Porter Stemming Algorithm (Jun. 2014) Disponível em: <http://tartarus.org/~martin/PorterStemmer/>
31. L. Wallace and M. Keil. “Software Project Risk and their Effect on Outcomes”, Communication fo the ACM, vol 47 número 4, pp. 68-73, 2004.
32. B.W. Boehm. “Software Risk Management: Principles and Practices”, IEEE Software vol 8, número 1, pp.32-41,1991
33. T. Addison. “E-Commerce Project Development Risks: Evidence from a Delphi survey”, International Journal of Information Management vol 23, number 1, pp. 25-40, 2003

34. IPQ (Instituto Português da Qualidade). Gestão do risco vocabulário DNP ISO Guia 73: 2011.
35. Prasanta Kumar, D. “Managing project risk using combined analytic hierarchy process and risk map.”, 2010
36. R.Flanagan e G.Norman. “Risk Management and Construction”, 1993
37. Auster Moreira Nascimento; Luciane Reginato. “Controladoria - um enfoque na eficácia organizacional”. São Paulo: Atlas, pp. 30-49, 2007
38. Projetosae5 (Jun. 2014) Disponível em: projetosae5.blogspot.pt/p/sad-sistemas-de-apoio-decisao.html
39. Luismarreiros (Jun. 2014) Disponível em: luismarreiros.blogspot.pt/2011/10/resenha-da-historia-dos-sistemas-de.html